

PATENT APPLICATION
Attorney Docket No.: SNS-009A (7268/15)

SYSTEMS AND METHODS FOR VOXEL WARPING

Cross-Reference to Related Applications

[0001] This application claims the benefit of U.S. provisional patent application Serial No. 60/255,530, filed December 14, 2000, which application is incorporated herein in its entirety by reference.

Field of the Invention

[0002] This invention relates generally to multi-dimensional modeling. More particularly, the invention relates to modeling using a change in the representation of the object to be modeled.

Background of the Invention

[0003] Traditional tools for computer-based editing of three-dimensional models are surface based. Often, a model is represented as a collection of surface elements such as polygons or NURBS. Such tools support aggregation, repositioning and deformation of models readily, but do not lend themselves to topological modifications such as punching holes. These models also do not readily provide a watertight model, that is, one that unambiguously separates inside from outside at every point in space. For example, a collection of surface elements can leave an unfilled hole in the model, or two surface elements can intersect in a way that makes inside and outside classification unclear.

Summary of the Invention

[0004] Volumetric representations readily maintain a watertight model without holes or self-intersections. Such representations naturally support Boolean operations based on a voxel-by-voxel compositing process. It is desirable to support deformations of voxel models to expand the range of operations that can be performed natively within the voxel grid with such models.

[0005] A method is described for local modification of voxel models. A voxel is a volume element. A voxel model is given initially. A region to be modified and a warping function are then specified. The inverse of the desired warping function is used to resample the

voxel grid over the region of interest. Associated surface or other display parameters are then updated.

[0006] In one aspect, the invention relates to a method of calculating a modification of a geometrical shape. The method comprises the steps of defining on a multi-dimensional space an array of values representing a geometrical shape; selecting a modification function that represents a desired modification to be applied to the geometrical shape; applying an inverse function of the modification function to the array of values to produce a modified array; and deducing from the modified array a modification of the geometrical shape that would result from a direct application of the modification function to the array.

[0007] In one embodiment, the method further comprises the step of applying the deduced modification to the array. In one embodiment, the method further comprises displaying to a user the modification of the geometrical shape that would result from a direct application of the modification function to the array. In one embodiment, the step of deducing from the modified array a modification further comprises the steps of retrieving an array value from the modified array, and applying the array value from the modified array at a location in the array representing the geometrical shape. In one embodiment, the modification function comprises a soft-edged deformation. In one embodiment, the modification function is applied from a selected one of a first side and a second side of the geometrical shape.

[0008] In one embodiment, the method further comprises applying the modification function by manipulation of a virtual tool. In one embodiment, the geometrical shape is displaced away from the virtual tool. In one embodiment, the geometrical shape is displaced toward the virtual tool. In one embodiment, the modification of the geometrical shape is substantially a convex hump. In one embodiment, the modification of the geometrical shape is substantially a concave spiked protuberance.

In one embodiment, the modification function comprises a force field consistent with a tool of arbitrary shape. In one embodiment, the modification function comprises translational displacement. In one embodiment, the modification function comprises rotational displacement.

[0009] In one embodiment, the modification function comprises a selected one of a displacement function, a smoothing function, a warping function, a volumetric interference, an

areal interference, a result of a simulation, a data re-fitting, and a force. In one embodiment, the modification function is represented as a non-linear mathematical function.

In one embodiment, the method further comprises the steps of selecting a second modification function; applying an inverse function of the second modification function to the modified array to produce a twice-modified array; and deducing from the twice-modified array a modification of the geometrical shape that would result from an application of the modification function to the array followed by an application of the second modification function to the once-modified array.

[0010] In one embodiment, the modification comprises applying a constraint so as to control a magnitude of a change of the geometrical shape. In one embodiment, applying the constraint prevents at least one point of the geometrical shape from moving in response to the application of the inverse function of the modification function. In one embodiment, the modification comprises applying a surface texture to the geometrical shape.

[0011] In another aspect the invention features a system for calculating a modification of a geometrical shape. The system comprises a generation module that defines on a multi-dimensional space an array of values representing a geometrical shape; a selection module that provides a modification function representing a desired modification to be applied to the geometrical shape; a transformation module that applies an inverse function of the modification function to the array of values to produce a modified array; and a calculation module that deduces from the modified array a modification of the geometrical shape that would result from a direct application of the modification function to the array.

[0012] In one embodiment, the system further comprises a modification module that applies the deduced modification to the array. In one embodiment, the system further comprises a display module that displays to a user the modification of the geometrical shape that would result from a direct application of the modification function to the array.

[0013] In one embodiment, the calculation module further comprises a module that retrieves an array value from the modified array, and a module that applies the array value from the modified array at a location in the array representing the geometrical shape. In one embodiment, the modification function comprises a soft-edged deformation. In one embodiment, the modification function is applied from a selected one of a first side and a second side of the geometrical shape.

[0014] In one embodiment, the system further comprises a module that applies the modification function by manipulation of a virtual tool. In one embodiment, manipulation of a virtual tool displaces the geometrical shape away from the virtual tool. In one embodiment, manipulation of a virtual tool displaces the geometrical shape toward the virtual tool. In one embodiment, the modification of the geometrical shape is substantially a convex hump. In one embodiment, the modification of the geometrical shape is substantially a concave spiked protuberance. In one embodiment, the modification function comprises a force field consistent with a tool of arbitrary shape. In one embodiment, the modification function comprises translational displacement. In one embodiment, the modification function comprises rotational displacement. In one embodiment, the modification function comprises a selected one of a displacement function, a smoothing function, a warping function, a volumetric interference, an areal interference, a result of a simulation, a data re-fitting, and a force. In one embodiment, the modification function is represented as a non-linear mathematical function.

[0015] In one embodiment, the system further comprises a selection module that selects a second modification function; a transformation module that applies an inverse function of the second modification function to the modified array to produce a twice-modified array; and a calculation module that deduces from the twice-modified array a modification of the geometrical shape that would result from an application of the modification function to the array followed by an application of the second modification function to the once-modified array. In one embodiment, the modification comprises applying a constraint so as to control a magnitude of a change of the geometrical shape. In one embodiment, the module that applies the constraint prevents at least one point of the geometrical shape from moving in response to the application of the inverse function of the modification function. In one embodiment, the modification comprises applying a surface texture to the geometrical shape.

[0016] The foregoing and other objects, aspects, features, and advantages of the invention will become more apparent from the following description and from the claims.

Brief Description of the Drawings

[0017] The objects and features of the invention can be better understood with reference to the drawings described below, and the claims. The drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of the invention. In the drawings, like numerals are used to indicate like parts throughout the various views.

[0018] Figure 1 is an illustrative flow diagram showing the steps performed according to an embodiment of the invention;

[0019] Figures 2A through 2D show an illustrative example of the application of a modification function to a two dimensional space represented by an array of values, according to principles of the invention;

[0020] Figure 3 is a graph of an exemplary forward warping function, according to an embodiment of the invention;

[0021] Figure 4 is a graph of an inverse of warping function based on the exemplary warping function of Figure 3, according to an embodiment of the invention;

[0022] Figure 5 is a diagram that depicts the components of an illustrative warp, according to an embodiment of the invention;

[0023] Figure 6 is a diagram that depicts the actions that correspond to an illustrative warp, according to an embodiment of the invention;

[0024] Figure 7 is a diagram of the result of the illustrative warp on the original voxel region, according to an embodiment of the invention;

[0025] Figure 8 is a graphical user interface used to activate the interactive smoothing tool, according to an embodiment of the invention;

[0026] Figure 9 is a graphical user interface used to control the properties of the interactive smoothing tool, according to an embodiment of the invention;

[0027] Figure 10 is a graphical user interface used to invoke the Smudge tool, according to an embodiment of the invention;

[0028] Figure 11 is a graphical user interface used to control a mode of operation of the Smudge tool, according to an embodiment of the invention;

[0029] Figure 12 is a graphical user interface used to control the properties of the Smudge tool, according to an embodiment of the invention;

[0030] Figure 13 is another graphical user interface used to control the properties of the Smudge tool, according to an embodiment of the invention;

[0031] Figure 14 is a graphical user interface used to control the properties of the Boolean smoothing tool, according to an embodiment of the invention;

5 [0032] Figure 15 is another graphical user interface used to control the properties of the Boolean smoothing tool, according to an embodiment of the invention;

[0033] Figure 16 is a graphical user interface pull-down menu used to activate the Attract and Spikes tools, according to an embodiment of the invention;

[0034] Figure 17 is a graphical user interface used to used to control the properties of the Attract tool, according to an embodiment of the invention;

[0035] Figure 18 is a graphical user interface used to used to control the properties of the Spikes tool, according to an embodiment of the invention;

[0036] Figure 19 is a graphical user interface used to depict the location of a tool in virtual space for the convenience of a viewer, according to an embodiment of the invention; and

[0037] Figures 20A through 20F are schematic diagrams showing two tool shape, the result of scaling the two tool shapes, and the results of offsetting the tools, according to an embodiment of the invention.

Detailed Description

20 [0038] The methods and systems of the invention are carried out using a conventional general purpose computer, such as a commercially available personal computer that comprises a CPU, one or more memories, one or more storage medium, a display, and an input device. The computer is programmed with software comprising commands that when operating direct the computer in the performance of the methods of the invention. Those of skill in the programming
25 arts will recognize that some or all of the commands can be provided in the form of software, in the form of programmable hardware such as flash memory or ROM, in the form of hard-wired circuitry, or in some combination of two or more of software, programmed hardware, or hard-wired circuitry. Commands that control the operation of a computer are often grouped into units that perform a particular action, such as receiving information, processing information or data,
30 and providing information to a user. Such a unit can comprise any number of instructions, from

a single command, such as a single machine language instruction, to a plurality of commands, such as a plurality of lines of code written in a higher level programming language such as C++. Such units of commands will be referred to generally as modules, whether the commands comprise software, programmed hardware or hard-wired circuitry, or a combination thereof.

5 [0039] In alternative embodiments, the computer is a laptop computer, a minicomputer, a mainframe computer, an embedded computer, or a handheld computer. The memory is any conventional memory such as, but not limited to, semiconductor memory, optical memory, or magnetic memory. The storage medium is any conventional machine-readable storage medium such as, but not limited to, floppy disk, hard disk, CD-ROM, and/or magnetic tape. The display is any conventional display such as, but not limited to, a video monitor, a printer, a speaker, an alphanumeric display and/or a force feedback haptic interface device. The input device is any conventional input device such as, but not limited to, a keyboard, a mouse, a force-feedback haptic interface device, a touch screen, a microphone, and/or a remote control. The computer can be a stand-alone computer or interconnected with at least one other computer by way of a network.

10017031521401
20 [0040] Figure 1 shows a flow diagram that illustrates the steps in one embodiment of the invention, and equivalently, illustrates the functions of modules that control the operation of the system. As indicated in box 102, a geometrical shape to be modified is defined as an array of values in a computer memory. As indicated in box 104, a user or a programmed computer selects a modification function to be applied to the geometrical shape. As indicated in box 106, the programmed computer applies an inverse function of the modification function to the array that represents the geometrical shape to be modified, which results in a modified array of values. As indicated at box 108, the programmed computer deduces a modification of the geometrical shape from the modified array of values. In some embodiments, as indicated at box 110, the deduced modification is optionally applied to the array of values in memory that represent the geometrical shape, and is available for storage for later use. In some embodiments, as indicated in box 112, the modification of the geometrical shape is optionally displayed to a user by way of a display device. The optional display of the modification of the geometrical shape can be performed independent of the optional application of the deduced modification to the array of values, as indicated by an arrow pointing directly from box 108 to box 112.

30

[0041] Figure 1 further shows that, in some embodiments, a second modification function can be applied to the modified array to further modify the geometrical shape. As indicated at box 114, in one embodiment, a user or a programmed computer selects a second modification function to be applied to the once-modified array corresponding to a once modified geometrical shape. As indicated at box 116, the programmed computer applies an inverse function of the second modification function to the modified array to produce a twice-modified array. As indicated at box 118, the programmed computer deduces from the twice-modified array a modification of the geometrical shape that would result from an application of the modification function to the array followed by an application of the second modification function to the once-modified array. In principle, any number of modifications can be performed by applying successive modification functions to an array. In addition, a modification can be undone by recalling a previously stored array corresponding to the geometrical shape as it was prior to the application of the modification function.

[0042] Figures 2A through 2D show an illustrative example of the application of a modification function to a two dimensional space represented by an array of values. The use of the inverse function is illustrated in the sequence of Figures 2A – 2D.

[0043] Figure 2A shows an initial condition of a two-dimensional object 200, which is, for example, a topographical map of an area. The values entered in the two-dimensional array of Figure 2A represent an elevation of the square upon which each value is displayed. The solid line 202 shows the locus of points having an elevation of 1.5 units, e.g., an isoelevation. The dotted line 204 encloses a region of the two-dimensional object 200 that is to be displaced one unit to the right. According to methods of the invention, rather than moving the region enclosed by dotted line 204 directly, an inverse function is determined, namely, moving the enclosing line 204 one unit to the left, and moving the array entries so identified one unit to the right.

[0044] In Figure 2B, the dotted line 204' illustrates the position of dotted line 204 after the inverse function is applied to dotted line 204. In Figure 2B, no array entry has yet been moved.

[0045] In Figure 2C, the array entries identified within the region circumscribed by the dotted line 204' of Figure 2B are copied to the locations identified by the original dotted line 204 of Figure 2A. It is to be observed that the net result of the operations illustrated in Figures 2B

and 2C is equivalent to moving each array entry found within the dotted line 204 of Figure 2A one unit to the right within Figure 2A.

[0046] Finally, in Figure 2D the line 202 is redrawn to indicate substantially the new locus of the isoelevation having a height of 1.5 units. It will be understood that this example is illustrative, and that appreciably more complex modifications of three-dimensional shapes are contemplated.

[0047] In order to perform the methods of the invention, an initial model is created. In a preferred embodiment, a voxel model is the input for the warping process. This model comes from various sources including scans of physical objects or prior interactive editing. The voxel model, or an alternative initial model, is represented by numerical values maintained in computer memory in an array.

[0048] A modification function used to modify the initial model is selected or otherwise identified by a user or by a programmed computer. The modification function is intended to be applied to the initial model in order to modify an attribute or value of the model, which represents a change in a geometrical shape of the object represented by the model.

[0049] A modification, or warping, function is any mathematical function that takes a multi-dimensional (commonly three-dimensional) position within some domain as an input and provides a multi-dimensional (commonly three-dimensional) position within some range as an output. An example is depicted in Figure 3. The modification or warping function 310 is preferably invertible, that is, for each point in the range a unique point in the domain can be determined. An inverted modification function 410 or warping function 410 based on the example in Figure 3 is depicted in Figure 4.

[0050] A modification or warping function can have constraints. In some embodiments, the modification or warping function will have a limited region of influence, and will have a magnitude of effect that falls to 0 (no displacement) at least one limit of the modification or warping function. These constraints prevent the introduction of discontinuities in the model.

[0051] In some embodiments, the domain and range are the same, for example, in the illustrative examples presented here. When the domain and the range differ, material moved in the course of the modification or warp can overlap previously existing material. When such an overlap occurs, a computation is performed to account for the overlap. In one illustrative

embodiment, the value that is accepted as the correct value is the maximum voxel value of one of the modified or warped voxel and the original voxel. In some embodiments, the modification that is needed is deduced by retrieving an array value from the array after it is modified, and applying the array value from the modified array at a location in the array representing the geometrical shape.

[0052] A smoothness constraint is useful to avoid introducing creases or other artifacts during modification or warping,. Such a constraint can be introduced by limiting a derivative of a tangent to a surface or an absolute change in magnitude of a surface value. In some embodiments, a constraint is applied so as to control a magnitude of a change in the geometrical shape. In some embodiments, the constraint prevents a point of the geometrical shape from moving.

[0053] In some embodiments, the modification is a surface texture that is applied to the geometrical shape. For example, a surface texture such as the "pebble grain" on the surface of a football, or a surface texture such as the indentations present on the surface of a golf ball, can be applied to a surface. In some embodiments, the modification is a soft-edged deformation, e.g. a deformation having a smooth and flowing shape at an edge. In some embodiments, the modification is applied from one side of a geometrical shape (for example, the inside). In other embodiments, the modification is applied from a second side of a geometrical shape (for example, the outside).

[0054] In some embodiments, the modification is accomplished by use of a modification function comprising a force field that is substantially consistent with a tool of arbitrary shape. In some embodiments, the modification function comprises translational displacement. In other embodiments, the modification function comprises rotational displacement. In some embodiments, the modification function comprises a selected one of a displacement function, a smoothing function, a warping function, a volumetric interference, an areal interference, a result of a simulation, a data re-fitting and a force. In some embodiments, the modification function is represented as a non-linear mathematical function.

[0055] Figure 3 is a graph of an exemplary forward warping function 310. The curve 310 of Figure 3 falls below a straight line rising at 45 degrees from the origin. One observes that under the influence of the function 310 depicted in Figure 3, material within the affected area

shrinks toward the center of projection if the force between the center of projection and the material comprising the geometrical shape is attractive. A center of projection is represented by a virtual tool displayed on a computer system display for a user. In other embodiments, different warping functions will have other properties. In some embodiments, the force between a center of projection and the geometrical shape is repulsive. In other embodiments, different warping functions will have other properties.

[0056] Figure 4 is a graph of an inverse warping function 410 based on the exemplary warping function 310 of Figure 3. The voxel values in the geometrical shape to be modified are computed by using the inverse function 410 to map output grid points in the array to input locations in the geometrical shape.

[0057] Figure 5 is a diagram that depicts the components of an illustrative warp. A geometrical shape 505 that to be modified is shown. In Figure 5, region A 510 depicts voxels that correspond to a scratch buffer region that is used for computation of the change in the geometrical shape. Region B 515 depicts a region of influence of the warping function, based on a central point P_0 520. Region C 525 represents the original voxel model of geometrical shape 505. For the selected modification function, only that portion of the model contained within the region of influence B 515 of the modification is expected to change shape.

[0058] The user or the programmed computer specifies a region of application for the warp. The portion of the voxel model to be altered by the warping function is specified, either through user interaction or automatically. This portion includes the mathematical union of the domain, that is, the portion of the voxel grid serving as input to the warp, and the range of the warping, the portion of the voxel grid containing the positions after warping.

[0059] The programmed computer determines a voxel region affected by the warp. This region includes the entire region of application of the warp determined in the previous step, but can larger. A larger region can in some instances simplify computer processing. In one embodiment, an example for such a region, the voxel region may be an axis-aligned rectangular region of a size that contains the region of application, which region of application may for example be irregular in shape. Alternative examples of computational regions are regions that circumscribe the voxel region of interest having a regular volumetric shape, having symmetry of various degrees, or having other computationally-efficient attributes.

[0060] Figure 6 is a diagram that depicts the actions that correspond to an illustrative warp. Figure 6 depicts regions A 510' and B 515', as well as central point P₀ 520'. In Figure 6, the second voxel region is filled by application of the inverse warping function to the geometrical shape. The region denoted by A 510' represents voxels unchanged by the warp. The region
5 denoted by B 515' represents voxels changed by resampling, as depicted in a two-dimensional representation of a multi-dimensional space. The changed shape is incorporated back into the original geometrical shape, as indicated with respect to Figure 7.

[0061] The programmed computer resamples the voxels. A second voxel region, which is referred to as the scratch buffer, is defined with the same dimensions as the voxel region
10 affected by the warp. For each grid point in the second voxel region, the corresponding point in the first voxel region is found by application of the inverse warp function. This point is possibly not aligned with the voxel grid. A value is computed for this output point by interpolation of the input voxel grid. In one embodiment, the computation is a trilinear interpolation over the 8 voxel neighbors of the input point. The derived value or values are stored in the scratch buffer at the output point location. Repeating the calculation for every voxel in the second voxel region
15 determines the complete warp.

[0062] Figure 7 is a diagram of the result of the illustrative warp on the original voxel
20 region. In Figure 7, the region C 525 is substantially the same shape as it was before the application of the modification function or its inverse function. The region A 510 includes voxels which were not affected by the application of the modification or warping function, even though they are part of the multi-dimensional representation that is being modified. The portion of the geometrical shape within the region B 515 are modified in shape by application of the inverse of the modification or warping function, and that change in shape is reflected in the original array, corresponding to the original geometrical shape. The changes in shape are
25 obtained, in one embodiment, from the scratch buffer array of modified data.

[0063] The scratch buffer voxels are copied into the original voxel grid. Each point stored in the second voxel grid is moved into the original voxel grid to effect the changes in the model. The voxel model properties are updated to reflect the changed region. Haptic or graphic properties are updated as appropriate, and are optionally presented to a user. In one embodiment,

a voxel display handled via an isosurface is updated by updating the isosurface for the changed region.

[0064] Some exemplary applications of voxel warping are presented below. The applications are embodied in a software package available from SensAble Technologies, Inc., Woburn, Massachusetts, which package is called the FreeForm™ modeling system. The software package includes a number of tools, which are presented using graphical user interface technology. As will be understood by those of skill in the programming arts, equivalent commands can be implemented using other interfaces, such as text-based interfaces. While one specific embodiment is presented for each tool, it will be understood by those of ordinary skill in the computer programming arts that other embodiments are possible. These tools are:

Interactive Smoothing: smoothing is performed while a ball-shaped tool is passed over a surface of a three-dimensional object.

Smudge: material is pushed away from a tool while carving/adding/extruding a three-dimensional shape.

Boolean Smooth: a smoothing while carving operation for use with a three-dimensional object.

Attract: material is attracted to the center of a tool. This operation is useful for raising bumps or creating smooth valleys in a three dimensional object.

Spikes: material is attracted to the edge of the tool and is useful in creating spikes.

The software includes interactive smoothing controlled by the smoothing dynabar of a user interface; a Smudge tool with its own left-hand toolbar button that operates in carve and pull modes; Boolean smoothing as a user-accessible selection for all of the carving tools; and Attract and Spikes as specialized-use tools on a Tools->Special Effects pull-down menu.

Interactive Smoothing Tool

[0065] Interactive Smoothing is the default selection on the smoothing flyout (a context-dependent user interface component brought up when its corresponding button is pressed) graphical user interface as shown in Figure 8. Once the user presses the Smooth button, (as distinct from a button called Smooth Area) on the left-hand bar, the smooth dynabar appears.

The smooth dynabar is a graphical user interface used to control the properties of the interactive

smoothing tool, and is shown in Figure 9. The sliders 910, 920 for Smoothing have similar controls and ranges as Smooth Area. The Smoothing tool size is the same visually as the Smooth Area tool and affects the same area. For instance, smoothing a swath yields a similar result as painting a swath, and then activating a command “apply” to apply the action to the representation of the area in memory. Once the tool size and smooth level is adjusted by the user by controlling sliders 910 and 920, respectively, the tool size and smooth level are defined until reset for the Smoothing tool, as is the case with the Smooth Area tool.

Smudge Tool

[0066] The Smudge tool has an icon 1010 below the ball tool as shown in the graphical user interface depicted in Figure 10. In one embodiment, the Smudge tool has two modes, Smudge from Inside 1110 and Smudge from Outside 1120, which are shown in Figure 11. The Smudge tool works well with the ball tool, according to one preferred embodiment. When the user selects the smudge tool, a status bar appear, which is a graphical user interface control shown in Figure 12.

[0067] One mode of controlling the tool size of the Smudge tool is to operate slider 1210. Motion of the slider 1210 to the left decreases tool size and motion to the right increases tool size, according to one embodiment of the invention. One mode of controlling the hardness of the clay that the user manipulates is to operate slide 1220. Motion of the slider 1220 to the left increases clay hardness and motion to the right decreases clay hardness, according to one embodiment of the invention. Another method of selecting and controlling tool size and clay hardness is the same as within the standard carving mode including the “T” key to toggle, the +/- keys for changing tool size, and the number keys for changing clay softness. When the user presses the “>>” key 1230, the Area of Influence slider 1340 pops up as shown in Figure 13.

[0068] The Area of Influence slider 1340 changes the amount of material outside of the tools shape that is affected by the Smudge tool and Smudge process. The slider has a range from 125% to 200% of the tool size radius with the middle setting as the default. These size and level setting persist during the work session, but do not need to be written in the preferences file. The result is that these settings work exactly the same as the tool/clay settings in Carving Tool Mode. When the user presses the “<<” button 1350, the Area of Influence slider 1340 will be hidden

from view, but retains its setting. The ">>" (Advanced) button state is written to the user preferences. Once the user has activated it, it stays open until closed by the user.

Boolean Smoothing Tool/ Smooth while Carving

5 [0069] The Boolean Smoothing tool, which provides the functionality of Smooth while Carving is an option that is controlled by activating the check box 1410 on the cutting tools dynabar 1400, as shown in Figure 14. In action, Boolean smoothing combines the effects of first carving a region to be modified in a standard Boolean fashion, then applying a local smoothing operation to the region, giving the effect of a soft-edged carve. When the user checks the smooth
10 box, Boolean smoothing commences using parameters defined by the mid-points of the two sliders described below in conjunction with Figure 15. When the ">>" (Advanced) button 1420 is selected, the bar shown in Figure 15 appears. Figure 15 shows a control bar that has increased in size from that of Figure 14 to display controls for the smoothed region outside of the cutting tool including Area of Influence slider 1510 (which scales from 125% to 200% of the tool size) and Smooth Level slider 1520. When the user presses the "<<" button 1530, the Area of Influence area is hidden from view, but retains its settings. When the advanced controls are available, but smooth is not selected, these sliders gray out to provide a visual notification to the user that Boolean smooth is not active. These size and level settings persist during the work session, but are not written in the preferences file. The ">>" (Advanced) button 1420 state is
20 written to the user preferences file once the user activates the Advanced feature. The Advanced feature remains open until closed by the user.

Attract and Spikes

[0070] Attract and Spikes are tools in the Special Effects library. These tools are
25 accessible on the Tools-> Special Effects menu 1600 as shown in Figure 16. Figure 16 is a graphical user interface pull-down menu used to activate the Attract and Spikes tools, which is used in a manner similar to other well-known pull-down menus. When the menu selection Attract 1610 is activated, the Attract tool is active. When the menu selection Spikes 1620 is activated, the Spikes tool is active. The check mark 1630 adjacent to the Spike 1620 menu
30 selection indicates that the Spikes tool is active.

Attract Tool

[0071] When Attract is chosen, the control bar 1700 shown in Figure 17 appears. The bar provides two buttons: Bulge 1710 and Indent 1720. These features are the equivalent of Attract from Outside and Attract from Inside, respectively. In one embodiment, the user controls tool size using the Tool Size slider 1730 and controls clay hardness using the slider 1740. These selections work the same as they do in carving tools mode. Activating the ">>" (Advanced) button, not shown, causes the Area of Influence slider 1750 to appear. The slider has a range of 125% to 200% of the tool size. The default is in the middle of the range. As with all of the other tools above, when the user presses the "<<" button 1760, the Area of Influence slider 1750 is hidden, but retains its setting. These size and level settings persist during the work session, but do not need to be written in the preferences file. The ">>" (Advanced) button state is written to the user preferences once the user has activated it. The Advanced button remains stay open until closed by the user. As with the other, similar toolbars, the "T" key toggles between the tool choices.

Spikes Tool

[0072] Figure 18 shows an exemplary graphical user interface 1800 for Spikes. This graphical user interface is nearly identical to the user interface for Attract, and works in a similar manner. The graphical user interface includes a Tool Size slider control 1810, a clay hardness slider control 1820, an area of influence slider control 1830 and a "<<" button 1840 that, when activated, hides the area of influence slider control 1830.

Tool Appearance

[0073] When any tool is selected that has an area of influence slider as described above, a semi-transparent tool shape 1900, of the appropriate size and shape, such as the embodiment shown in Figure 19, appears centered about the location of the tool in virtual space. This shape 1900 changes size in direct relationship with the Area of Influence Slider and with a nonlinear relationship with the Tool Size Slider, which represents a percentage, rather than an absolute value of the tool size. The semi-transparent shell 1910 disappears while the tool is modifying the

surface, for ease of viewer use. The transparent shell 1900 is offset from the cutting tool 1920, rather than being scaled, as is apparent from Figures 20A through 20F shown below. Figures 20A and 20D show two possible initial tool shapes, a polygon and an oval, respectively. Figures 20B and 20E show the effect of trying to offset the tool shape via simple scaling, giving in some cases an incorrect result. Figures 20C and 20F show the effect of offsetting the tool shape by a constant distance taking into account changes in shape during the offset. As shown in the examples, the offset oval is thicker than the original oval, and the offset polygon has different proportions of the corresponding edge lengths. In one embodiment a tool smoothes in a bounding box that fits around the tool. In another embodiment, the region matches the offset, rendered shell, rather than a bounding box.

INCORPORATION BY REFERENCE

[0074] United States patent application Serial Number 09/347,637, filed July 6, 1999, is hereby incorporated by reference herein in its entirety.

Equivalents

[0075] While the invention has been particularly shown and described with reference to specific preferred embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the appended claims.

[0076] What is claimed is: